

ICC - FUNÇÕES

PROF. FERNANDO W CRUZ

Roteiro da aula

- FUNÇÕES EM LINGUAGEM C
- PASSAGEM DE PARÂMETROS POR VALOR (no caso de variáveis simples) E POR REFERÊNCIA (no caso de variáveis dimensionais como vetores e matrizes).
- VARIÁVEIS LOCAIS E GLOBAIS
- EXERCÍCIOS

Discussão inicial

- Em linguagem C existem diversas funções predefinidas que podem ser utilizadas para construção de programas. Alguns exemplos: printf, scanf, random, resto da divisão ($m\%n$), ...
- Seria possível construir funções adicionais, além das que são nativas da linguagem C? Sim!

Algumas vantagens:

- i) Melhora o trabalho de abstração para construção de programas
- ii) Melhora a visualização da lógica dos programas
- iii) Permite o re-uso de rotinas muito utilizadas dentro de um programa

Exemplo I: Fatorial por somas sucessivas usando funções (I)

a) Elaborar um programa para calcular o fatorial de um número inteiro N positivo informado pelo usuário.

```
#include <stdio.h>
```

```
main() {
```

```
    int N, fat, i;
```

```
    system ("cls");
```

```
    printf("Digite o valor de N: ");
```

```
    scanf("%d", &N);
```

```
    i=N;
```

```
    fat = 1;
```

```
    while (i>0) {
```

```
        fat = fat * i;
```

```
        i = i-1;
```

```
    } /* fim-while */
```

```
    printf("Fatorial = %d\n", fat);
```

```
    system ("pause");
```

```
} /* fim-programa */
```

Chamada de uma função do sistema operacional (Windows) para limpar a tela antes de solicitar a digitação do valor de N

Perceba que, ao final da execução, a variável **fat** vai conter o fatorial de N.

Chamada de função do sistema operacional para evitar que a tela do programa suma imediatamente após a sua execução.

Exemplo I: Fatorial por somas sucessivas usando funções (II)

b) Elaborar um programa para calcular o produto de dois números inteiros pelo método das somas sucessivas.

```
#include <stdio.h>
```

```
main() {  
    int t1, t2, produto, contador;  
    printf("Digite o valor do primeiro termo: ");  
    scanf("%d", &t1);  
    printf("Digite o valor do segundo termo : ");  
    scanf("%d", &t2);
```

```
    contador= 0;  
    produto = 0;  
    while (contador < t2) {  
        produto = produto + t1;  
        contador = contador + 1;  
    } /* fim-while */
```

```
    printf("Produto de %d por %d = %d\n", t1, t2, produto);
```

```
} /* fim-programa */
```

Perceba que, ao final da execução, a variável **produto** fica com valor igual a **t1 * t2**

Exemplo I: Fatorial por somas sucessivas usando funções (II)

c) Elaborar um programa para calcular o fatorial de N substituindo a operação de multiplicação da questão anterior pelo método das somas sucessivas.

```
#include <stdio.h>

main() {
    int N, fat, i;
    int contador, produto;
    system("cls");
    printf("Digite o valor de N: ");
    scanf("%d", &N);
    i=N;  fat = 1;
    while (i>0) {
        contador = 0;
        produto = 0;
        while (contador < i) {
            produto = produto + fat;
            contador = contador + 1;
        } /* fim-while interno*/
        fat = produto;
        i = i-1;
    } /* fim-while externo*/
    printf("Fatorial = %d\n", fat);
} /* fim-programa */
```

Programa “produto por somas sucessivas” adaptado.
Perceba que:
i = t2 e que **fat = t1**
em relação ao programa do *slide* anterior.

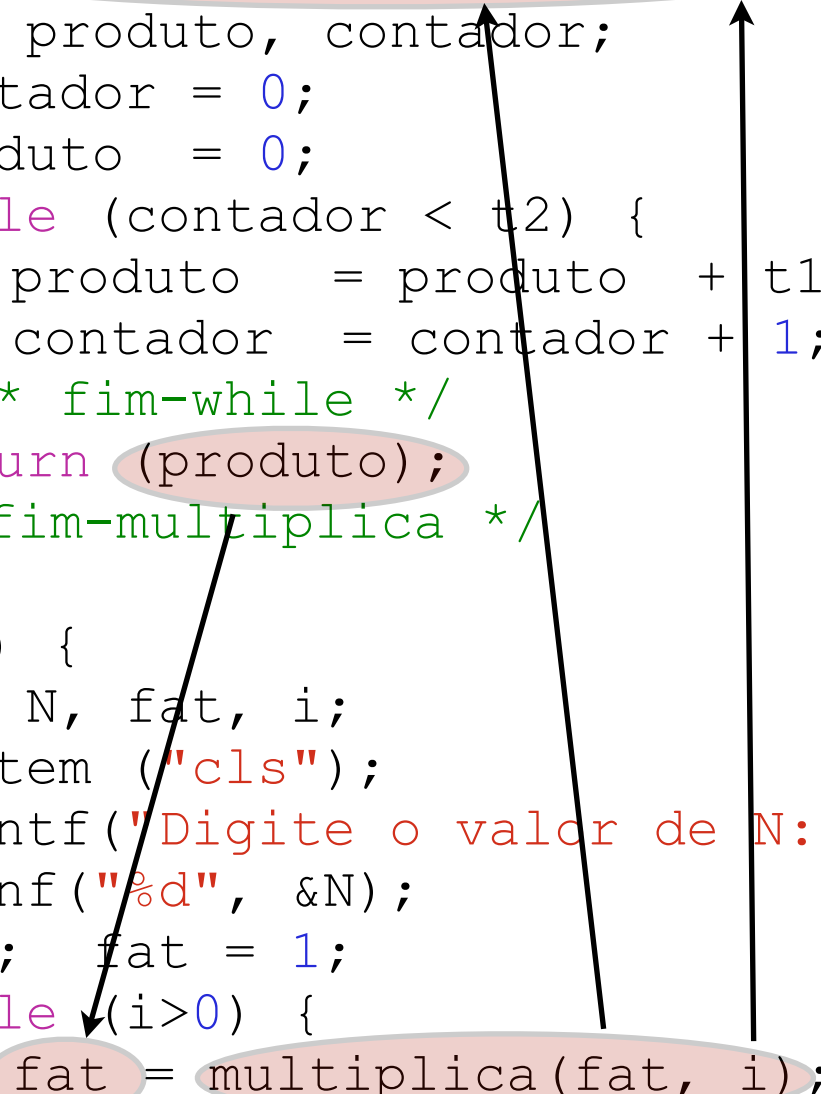
Exemplo I: Fatorial por somas sucessivas usando funções (IV)

d) Elaborar um programa para calcular o fatorial de N substituindo a operação de multiplicação da questão anterior por uma função que faça o cálculo pelo método das somas sucessivas.

```
#include <stdio.h>

int multiplica(int t1, int t2) {
    int produto, contador;
    contador = 0;
    produto = 0;
    while (contador < t2) {
        produto = produto + t1;
        contador = contador + 1;
    } /* fim-while */
    return (produto);
} /* fim-multiplica */

main() {
    int N, fat, i;
    system("cls");
    printf("Digite o valor de N: ");
    scanf("%d", &N);
    i=N; fat = 1;
    while(i>0) {
        fat = multiplica(fat, i);
        i = i-1;
    } /* fim-while */
    printf("Fatorial = %d\n", fat);
} /* fim-programa */
```



Observações:

- * A variável **fat** no ponto de chamada é igual a **t1** na função multiplica;
- * A variável **i** no ponto de chamada é igual a **t2** na função multiplica;
- * A passagem de parâmetros é por valor (t1 é cópia de fat e t2 é cópia de i)
- * As declarações de **t1** e **t2** são feitas dentro dos parênteses da função **multiplica**;
- * O comando **return** devolve ao ponto de chamada, o valor que está na variável produto. Nesse caso, a variável **fat** vai ter o mesmo valor da variável produto.

Exemplo II: Fatorial dos N termos de fibonacci usando funções (I)

a) Elaborar um programa para imprimir os N termos da série de Fibonacci.

```
#include <stdio.h>

main () {

    int ta, t1, t2, contador, N;

    system ("cls");
    printf("Qual o valor de N? ");
    scanf("%d", &N);
    t1 = 0;  t2 = 1;
    ta = t1+t2;
    contador = 0;
    while (contador < N) {

        printf("%d ", ta);

        ta = t1+t2;
        t1 = t2; t2 = ta;
        contador=contador + 1;
    } /*fim-while */
    printf("\n\n");
} /*fim-programa */
```

A variável **ta** contém o termo atual da série de Fibonacci.

Exemplo II: Fatorial dos N termos de fibonacci usando funções (II)

b) Elaborar um programa para imprimir o fatorial dos N termos da série de Fibonacci.

```
#include <stdio.h>
main () {
    int ta, t1, t2, contador, N;
    int fat, i;
    system ("cls");
    printf("Qual o valor de N? ");
    scanf("%d", &N);
    t1 = 0;  t2 = 1;
    ta = t1+t2;
    contador = 0;
    while (contador < N) {
        i = ta;
        fat = 1;
        while (i>0) {
            fat = fat*i;
            i = i-1;
        } /* fim-while */
        printf("%d(fat=%d) ", ta, fat);
        ta = t1+t2;
        t1 = t2; t2 = ta;
        contador = contador + 1;
    } /*fim-while */
    printf("\n\n");
} /*fim-programa */
```

A variável **ta** contém o termo atual da série de Fibonacci e a variável **fat** contém o fatorial de **ta**.

Exemplo II: Fatorial dos N termos de fibonacci usando funções (III)

c) Elaborar um programa para imprimir o fatorial dos N termos da série de Fibonacci, usando função para calcular os termos da série.

```
#include <stdio.h>
int calcula_fatorial(int i) {
    int fat;

    fat = 1;
    while (i>0) {
        fat = fat*i;
        i = i-1;
    } /* fim-while */
    return (fat);
} /* fim-calcula_fatorial */

main () {
    int ta, t1, t2, contador, N;
    system ("cls");
    printf("Qual o valor de N? ");
    scanf("%d", &N);
    t1 = 0; t2 = 1;
    ta = t1 + t2;
    contador = 0;
    while (contador < N) {
        printf("%d(fat=%d) ", ta, calcula_fatorial(ta));
        ta = t1 + t2;
        t1 = t2; t2 = ta;
        contador = contador +1;
    } /* fim-while */
    printf("\n\n");
} /*fim-programa */
```

* A variável **ta** no ponto de chamada é igual a **i** na função `calcula_fatorial` (elas precisam ser do mesmo tipo);

* O retorno da função `calcula_fatorial` substitui o nome da função onde ela é chamada;

* O comando **return** devolve ao ponto de chamada, o valor que está na variável produto. Nesse caso, a variável **fat** vai ter o mesmo valor da variável produto.

Exemplo II: Fatorial dos N termos de fibonacci usando funções (IV)

d) Elaborar um programa para imprimir o fatorial dos N termos da série de Fibonacci, usando função para calcular os termos da série e substituindo a função de multiplicação pelo método das somas sucessivas.

```
#include <stdio.h>
int multiplica (int t1, int t2) {
    int produto, contador;
    contador = 0; produto = 0;
    while (contador < t2) {
        produto = produto + t1;
        contador = contador + 1;
    } /* fim-while */
    return (produto);
} /*fim função multiplica */
```

```
int calcula_fatorial(int i) {
    int fat = 1;
    while (i>0) {
        fat = multiplica(fat,i);
        i = i-1;
    } /* fim-while */
    return (fat);
} /* fim-calcula_fatorial */
```

```
main () {
    int ta, t1, t2, contador, N;
    system ("clear"); printf("Qual o valor de N? ");
    scanf("%d", &N);
    t1 = 0; t2 = 1; ta = t1+t2;
    contador = 0;
    while (contador < N) {
        printf("%d(fat=%d) ", ta, calcula_fatorial(ta));
        ta = t1+t2; t1 = t2; t2 = ta;
        contador = contador +1;
    } /*fim-while */
    printf("\n\n");
} /*fim-programa */
```

Observações:

* Nesse caso, há uma hierarquia de chamada de funções:

main ⇨ *calcula_fatorial* ⇨ *multiplica*;

* Variáveis definidas dentro de uma função possuem valores que valem apenas dentro do contexto da função.

* Variáveis dentro de funções diferentes podem ter o mesmo nome, mas com conteúdos diferentes. Veja exemplo da variável **contador**.

* Se houver necessidade, pode-se criar variáveis globais (válidas para todas as funções). Nesse caso, devem ser declaradas fora das funções (abaixo da diretiva include).

* Variáveis multidimensionais também podem ser passadas como parâmetro de funções. (veja exercícios no próximo slide)

Exercício

Elaborar um programa para ler um vetor A de 10 posições. Em seguida calcular e imprimir o valor médio entre os elementos do vetor. Obs.: utilizar uma função para obter o valor médio entre os elementos do vetor.

Solução do exercício de vetor manipulado por uma função

```
#include <stdio.h>
```

```
float calc_media (int X[]) {
```

```
    int i = 0;
```

```
    float media;
```

```
    int soma = 0;
```

```
    while (i<10) {
```

```
        soma = soma + X[i];
```

```
        i=i+1;
```

```
    } /* fim-while */
```

```
    media = (float) soma/10;
```

```
    return (media);
```

```
} /* fim-calc_media */
```

```
main () {
```

```
    int A[10];
```

```
    float media;
```

```
    int i = 0;
```

```
    while (i<10) {
```

```
        printf("Digite A[%d]: ", i+1);
```

```
        scanf("%d", &A[i]);
```

```
        i=i+1;
```

```
    } /* fim-while */
```

```
    media = calc_media (A);
```

```
    printf("\nMedia de A = %.2f\n\n", media);
```

```
} /* fim-programa */
```

Observações:

* O vetor X recebe uma **referência** do vetor A. Perceba que, nesse caso, não precisa declarar o tamanho da variável X, já que ela refere-se à mesma posição de memória onde está o vetor A. Portanto, enquanto as variáveis simples podem ser passadas por valor (uma cópia da original), as variáveis dimensionais só podem ser passadas por referência (ou seja, A e X são *labels* que apontam para o mesmo local na memória).

* Nessa fórmula, o cálculo da média possui um *cast* (*float*) para garantir que a divisão da variável inteira *soma* por 10 vai retornar um valor real (ponto flutuante).

* Perceba que as variáveis declaradas são locais porque as declarações estão dentro do escopo de cada função. Por esse motivo, podem ter nomes similares, como é o caso da variável **media**. Se houvesse necessidade de uma variável global, a sua declaração poderia estar logo abaixo da linha **#include <stdio.h>**.

Exercício extra

Fazer um programa C para ler uma matriz A de 4 linhas e 4 colunas com números inteiros. Em seguida, criar uma função para descobrir e retornar a soma entre os elementos que estão abaixo da diagonal principal de A. O valor retornado deve ser impresso pelo programa principal.